Method for Transforming a Digital Signal from the Time Domain into the Frequency Domain and Vice Versa

Cross Reference to Related Applications

This application claims the benefit of priority of U.S. Provisional Application no 60/507,210, filed 29 September 2003, and U.S. Provisional Application no 60/507,440, filed 29 September 2003, the contents of each being hereby incorporated by reference in its entirety for all purposes.

Further, the following commonly-owned applications are concurrently-filed herewith, and herein incorporated in its entirety:

"Method for Performing a Domain Transformation of a Digital Signal from the Time Domain into the Frequency Domain and Vice Versa," Atty. Docket No. P100442, and

"Process and Device for Determining a Transforming Element for a Given Transformation Function, Method and Device for Transforming a Digital Signal from the Time Domain into the Frequency Domain and vice versa and Computer Readable Medium," Atty. Docket No. P100452.

Background

This invention relates to a method for transforming a digital signal from the time domain into the frequency domain and vice versa.

Domain transformations, for example the discrete cosine transform (DCT), are widely used in modern signal processing industry. Recently, a variant of the DCT, called integer DCT, has attracted a lot of research interests because of its important role in lossless coding applications. The term

"lossless" means that the decoder can generate an exact copy of the source signal from the encoded bit-stream.

The DCT is a real-valued block transform. Even if the input block consists only of integers, the output block of the DCT can comprise non-integer components. For convenience, the input block is referred to as input vector and the output block as output vector. If a vector comprises only integer components, it is called an integer vector. In contrast to the DCT, the integer DCT generates an integer output vector from an integer input vector. For the same integer input vector, the integer output vector of integer DCT closely approximates the real output vector of DCT. Thus the integer DCT keeps all the good properties of the DCT in spectrum analysis.

An important property of the integer DCT is reversibility. Reversibility means that there exists an integer inverse DCT (IDCT) so that if the integer DCT generates an output vector $\underline{y}$ from an input vector $\underline{x}$, the integer IDCT can recover the vector $\underline{x}$ from the vector $\underline{y}$. Sometimes the integer DCT is also referred to as the forward transform, and the integer IDCT as the backward or inverse transform.

A transform called integer modified discrete cosine transform (IntMDCT) is recently proposed and used in the ISO/IEC MPEG-4 audio compression. The IntMDCT can be derived from its prototype - the modified discrete cosine transform (MDCT). The disclosure by H. S. Malvar in "Signal Processing with Lapped Transforms" *Artech House*, 1992 provides an efficient realization of MDCT by cascading a bank of Givens rotations with a DCT-IV block. It is well known that Givens rotation can be factorised into three lifting steps for mapping

integers to integers. See e.g., R. Geiger, T. Sporer, J. Koller, K. Brandenburg, "Audio Coding based on Integer Transforms" *AES 111th Convention*, New York, USA, Sept. 2001.

Therefore, the realization of IntMDCT relies on an efficient implementation of integer DCT-IV.
Integer transforms can be directly converted from their prototypes by replacing each Givens rotation with three lifting steps. Because in each lifting step there is one rounding operation, the total rounding number of an integer transform is three times the Givens rotation number of the prototype transform. For discrete trigonometric transforms (for example the Discrete Fourier Transform (DFT) or the Discrete Cosine Transform (DCT)), the number of Givens rotations involved is usually at $N\log_2 N$ level, where N is the size of the blocks, i.e. the amount of data symbols included in each block, the digital signal is divided into. Accordingly, the total rounding number is also at $N\log_2 N$ level for the family of directly converted integer transforms. Because of the roundings, an integer transform only approximates its floating-point prototype. The approximation error increases with the number of roundings.

Accordingly what is needed are systems and methods for domain transforming a digital signal in a more efficient manner.

Summary of the Invention

The present invention provides systems and methods for domain transforming a digital signal, whereby two blocks of input data are concurrently domain transformed in the same operation. This configuration reduces the number of effective rounding operations, and accordingly the approximation error.

In one embodiment of the invention, a method of the invention a method for transforming a digital signal from the time domain into the frequency domain and vice versa using a transformation function is presented. The transformation function comprises a transformation matrix, the digital signal comprises data symbols which are grouped into a plurality of blocks, each block comprising a predefined number of the data symbols. The method comprises transforming two blocks of the digital signal by one transforming element, wherein the transforming element corresponds to a block-diagonal matrix comprising two sub-matrices, wherein each sub-matrices comprises the transformation matrix and the transforming element comprises a plurality of lifting stages and wherein each lifting stage comprises the processing of blocks of the digital signal by an auxiliary transformation and by a rounding unit.

These and other features of the invention will be better understood when viewed in light of the drawings and detailed description of the specific embodiments.

Brief Description of the Drawings

Figure 1 shows the architecture of an audio encoder according to an embodiment of the invention;

Figure 2 shows the architecture of an audio decoder according to an embodiment of the invention, which corresponds to the audio coder shown in figure 1;

Figure 3 shows a flow chart of an embodiment of the method according to the invention;

Figure 4 illustrates an embodiment of the method according to the invention using DCT-IV as the transformation function;

Figure 5 illustrates the algorithm for the reverse transformation according to the embodiment of the method according to the invention illustrated in figure 4;

Figure 6 shows the architecture for an image archiving system according to an embodiment of the invention;

Figure 7 shows forward and reverse transform coders used to evaluate the performance of the proposed system and method.

Detailed Description of Specific Embodiments

Figure 1 shows the architecture of an audio encoder 100 according to an embodiment of the invention. The audio encoder 100 comprises a conventional perceptual base layer coder based on the modified discrete cosine transform (MDCT) and a lossless enhancement coder based on the integer modified discrete cosine transform (IntMDCT).

An audio signal 109 which, for instance, is provided by a microphone 110 and which is digitalized by a analog-to-digital converter 111 is provided to the audio encoder 100. The audio signal 109 comprises a plurality of data symbols. The audio signal 109 is divided into a plurality of blocks, wherein each block comprises a plurality of data symbols of the digital signal and each block is transformed by a modified discrete cosine transform (MDCT) device 101. The MDCT coefficients are quantized by a quantizer 103 with the help of a perceptual model 102. The perceptual model controls the quantizer 103 in such a way that the audible distortions resulting from the quantization error are low. The quantized

MDCT coefficients are subsequently encoded by a bitstream encoder 104 which produces the lossy perceptually coded output bitstream 112.

The bitstream encoder 104 losslessly compresses its input to produce an output which has a lower average bit-rate than its input by standard methods such as Huffman-Coding or Run-Length-Coding. The input audio signal 109 is also fed into an IntMDCT device 105 which produces IntMDCT coefficients. The quantized MDCT coefficients, which are the output of the quantizer 103, are used to predict the IntMDCT coefficients. The quantized MDCT coefficients are fed into an inverse quantizer 106 and the output (restored or non-quantized MDCT coefficients) is fed into a rounding unit 107.

The rounding unit rounds to an integer value the supplied MDCT coefficients, and the residual IntMDCT coefficients, which are the difference between the integer value MDCT and the IntMDCT coefficients, are entropy coded by an entropy coder 108. The entropy encoder, analogous to the bitstream encoder 104, losslessly reduces the average bit-rate of its input and produces a lossless enhancement bitstream 113. The lossless enhancement bit stream 113 together with the perceptually coded bitstream 112, carries the necessary information to reconstruct the input audio signal 109 with minimal error.

Figure 2 shows the architecture of an audio decoder 200 comprising an embodiment of the invention, which corresponds to the audio coder 100 shown in figure 1. The perceptually coded bitstream 207 is supplied to a bitstream decoder 201, which performs the inverse operations to the operations of the bitstream encoder 104 of Figure 1, producing a decoded

bitstream. The decoded bitstream is supplied to an inverse quantizer 202, the output of which (restored MDCT coefficients) is supplied to the inverse MDCT device 203. Thus, the reconstructed perceptually coded audio signal 209 is obtained.

The lossless enhancement bitstream 208 is supplied to an entropy decoder 204, which performs the inverse operations to the operations of the entropy encoder 108 of figure 1 and which produces the corresponding residual IntMDCT coefficients. The output of the inverse quantizer 202 is rounded by a rounding device 205 to produce integer value MDCT coefficients. The integer value MDCT coefficients are added to the residual IntMDCT coefficients, thus producing the IntMDCT coefficients. Finally, the inverse IntMDCT is applied to the IntMDCT coefficients by an inverse IntMDCT device 206 to produce the reconstructed losslessly coded audio signal 210.

Figure 3 shows a flow chart 300 of an embodiment of the method according to the invention using the DCT-IV as a transformation and using three lifting stages, a first lifting stage 301, a second lifting stage 302 and a third lifting stage 303. This method is preferably used in the IntMDCT device 105 of figure 1 and the inverse IntMDCT device 206 of figure 2 to implement IntMDCT and inverse IntMDCT, respectively. In Fig.3, $\underline{x}_1$ and $\underline{x}_2$ are first and second blocks ~~of the digital~~ signal, respectively. $\underline{z}$ is an intermediate signal, and $\underline{y}_1$ and $\underline{y}_2$ are output signals corresponding to the ~~first~~ and second block of the digital signal, respectively.

As explained above, the DCT-IV-algorithm plays an important role in lossless audio coding.

The transformation function of the DCT-IV comprises the transformation matrix $\underline{C_N^{IV}}$. According to this embodiment of the invention, the transforming element corresponds to a block-diagonal matrix comprising two blocks, wherein each block comprises the transformation matrix $\underline{C_N^{IV}}$.

So, in this embodiment, the transformation matrix corresponding to the transforming element according to the invention is:

$$\begin{bmatrix} & \underline{C_N^{IV}} \\ \underline{C_N^{IV}} & \end{bmatrix}$$

$\underline{C_N^{IV}}$ shall in the context of this embodiment be referred to as the transformation matrix henceforth.

The number of lifting matrices, and hence the number of lifting stages in the transformation element, in this embodiment of the invention, wherein DCT-IV is the transformation function, is three.

The DCT-IV of an N-point real input sequence $x(n)$ is defined as follows:

$$y(m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos\left(\frac{(m+1/2)(n+1/2)\pi}{N} \right) \quad m,n = 0,1,\cdots,N-1 \qquad (1)$$

Let $\underline{C_N^{IV}}$ be the transformation matrix of DCT-IV, that is

$$\underline{C_N^{IV}} = \sqrt{\frac{2}{N}}\left[\cos\left(\frac{(m+1/2)(n+1/2)\pi}{N}\right)\right]_{m,n=0,1,\cdots,N-1} \tag{2}$$

The following relation holds for the inverse DCT-IV matrix:

$$\left(\underline{C_N^{IV}}\right)^{-1} = \underline{C_N^{IV}} \tag{3}$$

In particular, the matrix $\underline{C_N^{IV}}$ is involutory.

With $\underline{x} = [x(n)]_{n=0,1,\cdots,N-1}$ and $\underline{y} = [y(m)]_{m=0,1,\cdots,N-1}$, equation (1) can be expressed as

$$\underline{y} = \underline{C_N^{IV}}\,\underline{x} \tag{4}$$

Now, let $\underline{x_1}$, $\underline{x_2}$ be two integer $N\times 1$ column vectors. The column vectors $\underline{x_1}$, $\underline{x_2}$ correspond to two blocks of the digital signal which, according to the invention, are transformed by one transforming element. The DCT-IV transforms of $\underline{x_1}$, $\underline{x_2}$ are $\underline{y_1}$, $\underline{y_2}$, respectively.

$$\underline{y_1} = \underline{C_N^{IV}\,x_1} \tag{5}$$

$$\underline{y_2} = \underline{C_N^{IV}\,x_2} \tag{6}$$

Combining (5) and (6):

$$\begin{bmatrix}\underline{y_1}\\ \underline{y_2}\end{bmatrix} = \begin{bmatrix}\underline{C_N^{IV}} & \\ & \underline{C_N^{IV}}\end{bmatrix}\begin{bmatrix}\underline{x_1}\\ \underline{x_2}\end{bmatrix} \tag{7}$$

The above diagonal matrix is the block-diagonal matrix that the transforming element, according to the invention, corresponds to.

It is within the scope of the invention if the above equation is changed by simple algebraic modifications like the one leading to

$$\begin{bmatrix} \underline{y_1} \\ \underline{y_2} \end{bmatrix} = \begin{bmatrix} & \underline{C_N^{IV}} \\ \underline{C_N^{IV}} & \end{bmatrix} \begin{bmatrix} \underline{x_2} \\ \underline{x_1} \end{bmatrix} \tag{8}$$

Let $\underline{T_{2N}}$ be the counter diagonal matrix in (8), that is

$$\underline{T_{2N}} = \begin{bmatrix} & \underline{C_N^{IV}} \\ \underline{C_N^{IV}} & \end{bmatrix} \tag{9}$$

The matrix $\underline{T_{2N}}$ can be factorised as follows:

$$\underline{T_{2N}} = \begin{bmatrix} & \underline{C_N^{IV}} \\ \underline{C_N^{IV}} & \end{bmatrix} = \begin{bmatrix} \underline{I_N} & \\ -\underline{C_N^{IV}} & \underline{I_N} \end{bmatrix} \begin{bmatrix} -\underline{I_N} & \underline{C_N^{IV}} \\ & \underline{I_N} \end{bmatrix} \begin{bmatrix} \underline{I_N} & \\ \underline{C_N^{IV}} & \underline{I_N} \end{bmatrix} \tag{10}$$

where $\underline{I_N}$ is the $N \times N$ identity matrix.

Equation (10) can be easily verified using the DCT-IV property in (3). Using (10), Equation (8) can be expressed as

$$\begin{bmatrix} \underline{y_1} \\ \underline{y_2} \end{bmatrix} = \begin{bmatrix} \underline{I_N} & \\ -\underline{C_N^{IV}} & \underline{I_N} \end{bmatrix} \begin{bmatrix} -\underline{I_N} & \underline{C_N^{IV}} \\ & \underline{I_N} \end{bmatrix} \begin{bmatrix} \underline{I_N} & \\ \underline{C_N^{IV}} & \underline{I_N} \end{bmatrix} \begin{bmatrix} \underline{x_2} \\ \underline{x_1} \end{bmatrix} \tag{11}$$

The three lifting matrices in equation (11) correspond to the three lifting stages shown in Fig. 3.

From (11), the following integer DCT-IV algorithm that computes two integer DCT-IVs with one transforming element is derived.

Figure 4 illustrates the embodiment of the method according to the invention using DCT-IV as the transformation function. This embodiment is used in the audio coder 100 shown in Fig. 1 for implementing IntMDCT. Like in Figure 3, $\underline{x_1}$ and $\underline{x_2}$ are two blocks of the input digital signal, $\underline{z}$ is an intermediate signal, and $\underline{y_1}$ and $\underline{y_2}$ are corresponding blocks of the output signal.

The three lifting stages illustrated in figure 4 correspond to the three lifting matrices in equation (11).

As illustrated by figure 4, the time to frequency domain integer transform is determined by the following:

In the first stage 401, $\underline{x_2}$ is transformed by a DCT-IV transformation 402 and the DCT-IV coefficients are rounded 403. The rounded DCT-IV coefficients are then added to $\underline{x_1}$ 404. Thus, the intermediate signal $\underline{z}$ is generated. So, the intermediate signal $\underline{z}$ fulfils the equation:

$$\underline{z} = \left\lfloor C_N^{IV} \underline{x_2} \right\rfloor + \underline{x_1} \quad (12a)$$

In the second stage 405, $\underline{z}$ is transformed by a DCT-IV transformation 406 and the DCT-IV coefficients are rounded 407. From the rounded DCT-IV coefficients $\underline{x_1}$ is then subtracted. Thus, the output signal $\underline{y_1}$ is generated. So, the output signal $\underline{y_1}$ fulfils the equation:

$$\underline{y_1} = \left\lfloor C_N^{IV} \underline{z} \right\rfloor - \underline{x_2} \tag{12b}$$

In the third stage 409, $\underline{y_1}$ is transformed by a DCT-IV transformation 410 and the DCT-IV coefficients are rounded 411. The rounded DCT-IV coefficients are then subtracted from $\underline{z}$. Thus, the output signal $\underline{y_2}$ is generated. So, the output signal $\underline{y_2}$ fulfils the equation:

$$\underline{y_2} = -\left\lfloor C_N^{IV} \underline{y_1} \right\rfloor + \underline{z} \tag{12c}$$

where $\lfloor * \rfloor$ denotes rounding operation.

Figure 5 illustrates the algorithm for the reverse transformation according to an embodiment of the method according to the invention using DCT-IV as the transformation function. This embodiment is used in the audio decoder 200 shown in Fig. 2 for implementing inverse IntMDCT. The algorithm illustrated in figure 5 is the inverse of the algorithm illustrated in figure 4. The denotations for the different signals $\underline{y_1}$, $\underline{y_2}$, $\underline{x_1}$, $\underline{x_2}$ and $\underline{z}$ are chosen corresponding to the denotations of figure 4.

As illustrated by Figure 5, the frequency to time domain integer transform is determined by the following:

In the first stage 501, $\underline{y_1}$ is transformed by a DCT-IV transformation 502 and the DCT-IV coefficients are rounded 503. The rounded DCT-IV coefficients are then added to $\underline{y_2}$ 504. Thus, the intermediate signal $\underline{z}$ is generated. So, the intermediate signal $\underline{z}$ fulfils the equation:

$$\underline{z} = \left\lfloor C_N^{IV} \underline{y_1} \right\rfloor + \underline{y_2} \tag{13a}$$

In the second stage 505, $\underline{z}$ is transformed by a DCT-IV transformation 506 and the DCT-IV coefficients are rounded 507. From the rounded DCT-IV coefficients $\underline{y_1}$ is then subtracted. Thus, the signal $\underline{x_2}$ is generated. So, the signal $\underline{x_2}$ fulfils the equation:

$$\underline{x_2} = \lfloor C_N^{IV} \underline{z} \rfloor - \underline{y_1} \qquad (13b)$$

In the third stage 509, $\underline{x_2}$ is transformed by a DCT-IV transformation 510 and the DCT-IV coefficients are rounded 511. The rounded DCT-IV coefficients are then subtracted from $\underline{z}$. Thus, the signal $\underline{x_1}$ is generated. So, the signal $\underline{x_1}$ fulfils the equation:

$$\underline{x_1} = -\lfloor C_N^{IV} \underline{x_2} \rfloor + \underline{z} \qquad (13c)$$

It can be seen that the algorithm according to the equations (13a) to (13c) is inverse to the algorithm according to the equations (12a) to (12c). Thus, if used in the encoder and decoder illustrated in figures 1 and 2, the algorithms provide a method and an apparatus for lossless audio coding.

In an embodiment of the invention explained below, the method described above is used for an image archiving system.

The equations (12a) to (12c) and (13a) to (13c) further show that to compute two $N \times N$ integer DCT-IVs, three $N \times N$ DCT-IVs, three $N \times 1$ roundings, and three $N \times 1$ additions are needed. Therefore, for one $N \times N$ integer DCT-IV, the average is:

$$RC(N) = 1.5N \tag{14}$$

$$AC(N) = 1.5AC(C_N^{IV}) + 1.5N \tag{15}$$

where $RC(.)$ is the total rounding number, and $AC(.)$ is the total number of arithmetic operations. Compared to the directly converted integer DCT-IV algorithms, the proposed integer DCT-IV algorithm reduces RC from level $N\log_2 N$ to N.

As indicated by (15), the arithmetic complexity of the proposed integer DCT-IV algorithm is about 50 percent more than that of a DCT-IV algorithm. However, if RC is also considered, the combined complexity (AC+RC) of the proposed algorithm does not much exceed that of the directly converted integer algorithms. Exact analysis of the algorithm complexity depends on the DCT-IV algorithm used.

As shown in figures 4 and 5, the proposed integer DCT-IV algorithm is simple and modular in structure. It can use any existing DCT-IV algorithms in its DCT-IV computation block. The proposed algorithm is suitable for applications that require IntMDCT, e.g. in the MPEG-4 audio extension 3 reference model 0.

Figure 6 shows the architecture of an image archiving system according to an embodiment of the invention.

In Figure 6 an image source 601, for instance a camera, provides an analog image signal. The image signal is processed by a analog-to-digital converter 602 to provide a corresponding digital image signal. The digital image signal is losslessly encoded by a lossless image-encoder 603 which includes a transformation from the time domain to the

frequency domain. In this embodiment, the time domain corresponds to the coordinate space of the image. The lossless coded image signal is stored in a storage device 604, for example a hard disk or a DVD. When the image is needed, the losslessly coded image signal is fetched from the storage device 604 and provided to a lossless image decoder 605 corresponding to the lossless image encoder 603 which decodes the losslessly coded image signal and reconstructs the original image signal without any data loss.

Such lossless archiving of image signals is important, for example, in the case that the images are error maps of semiconductor wafers and have to be stored for later analysis.

In this embodiment of the invention, the embodiment of the method illustrated in figures 3 to 5 is used in the lossless image encoder 603 and the lossless image decoder 605. As explained above, the embodiment of the method illustrated in figures 3 to 5 provides a transformation which is reversible, thus in particular providing a method for lossless image coding.

The method according to the invention is not limited to audio are image signals. Other digital signals, for example video signals, can as well be transformed by the method according to the invention.

In the following, a further embodiment of the method for the transformation of a digital signal from the time domain to the frequency domain and vice versa according to the invention is explained.

In this embodiment of the present invention, the domain transformation is a DCT transform, whereby the block size N is some integer. In one embodiment, N is a power of two.

Let $\mathbf{C}_N^{II}$ be the $N \times N$ transform matrix of DCT (also called Type-II DCT):

$$\mathbf{C}_N^{II} = \sqrt{2/N}\left[k_m \cos\left(m\left(n+1/2\right)\pi/N\right)\right] \quad m,n = 0,1,\cdots,N-1 \tag{16}$$

where:

$$k_m = \begin{cases} 1/\sqrt{2} & \text{if} \quad m = 0 \\ 1 & \text{if} \quad m \neq 0 \end{cases} \tag{17}$$

and N is the transform size. m and n are matrix indices.

Let $\mathbf{C}_N^{IV}$ be the $N \times N$ transform matrix of type-IV DCT, as already defined above:

$$\mathbf{C}_N^{IV} = \sqrt{2/N}\left[\cos\left(\left(m+1/2\right)\left(n+1/2\right)\pi/N\right)\right] \quad m,n = 0,1,\cdots,N-1 \tag{18}$$

As above, a a plurality of lifting matrices will be used, which lifting matrices are in this embodiment $2N \times 2N$ matrices of the following form:

$$\mathbf{L}_{2N} = \begin{bmatrix} \pm\mathbf{I}_N & \mathbf{A}_N \\ \mathbf{O}_N & \pm\mathbf{I}_N \end{bmatrix} \tag{19}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix, $\mathbf{O}_N$ is the $N \times N$ zero matrix, $\mathbf{A}_N$ is an arbitrary $N \times N$ matrix.

For each lifting matrix $\mathbf{L}_{2N}$, a lifting stage reversible integer to integer mapping is realized in the same way as the 2x2 lifting step described in the incorporated reference "Factoring Wavlet Transforms into Lifting Steps," *Tech. Report*, I. Daubechies and W. Sweldens, Bell Laboratories, Lucent Technologies, 1996. The only difference is that rounding is applied to a vector instead of a single variable.

In the above description of the other embodiments, it was already detailed how a lifting stage is realized for a lifting matrix, so the explanation of the lifting stages corresponding to the lifting matrices will be omitted in the following.

One sees that the transposition of $\mathbf{L}_{2N}$, $\mathbf{L}_{2N}^T$ is also a lifting matrix.

In this embodiment, the transforming element corresponds to a matrix, $\mathbf{T}_{2N}$ which is defined as a $2N \times 2N$ matrix in the following way:

$$\mathbf{T}_{2N} = \begin{bmatrix} \mathbf{C}_N^{IV} & \mathbf{O}_N \\ \mathbf{O}_N & \mathbf{C}_N^{IV} \end{bmatrix} \quad (20)$$

The decomposition of the matrix $\mathbf{T}_{2N}$ into lifting matrices has the following form:

$$\mathbf{T}_{2N} = \mathbf{P3} \cdot \mathbf{L8} \cdot \mathbf{L7} \cdot \mathbf{L6} \cdot \mathbf{P2} \cdot \mathbf{L5} \cdot \mathbf{L4} \cdot \mathbf{L3} \cdot \mathbf{L2} \cdot \mathbf{L1} \cdot \mathbf{P1} \quad (21)$$

The matrices constituting the right hand side of the above equation will be explained in the following.

P1 is a first permutation matrix given by the equation

$$\mathbf{P1} = \begin{bmatrix} \mathbf{O}_N & \mathbf{D}_N \\ \mathbf{J}_N & \mathbf{O}_N \end{bmatrix} \tag{22}$$

where $\mathbf{J}_N$ is the $N \times N$ counter index matrix given by

$$\mathbf{J}_N = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & \iddots & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{23}$$

and $\mathbf{D}_N$ is a $N \times N$ diagonal matrix with diagonal element being 1 and -1 alternatively:

$$\mathbf{D}_N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{24}$$

P2 is a second permutation matrix, an example of which is generated by the following MATLAB script:

```
=======================================================
   Pd = eye(2*N);
   for i = 2:2:N,
       Pd(i,i) = 0; Pd(N+i,N+i) = 0;
       Pd(i,N+i) = 1; Pd(N+i,i) = 1;
   end
```

```
Peo = zeros(2*N);
for i = 1:N,
    Peo(i,2*i-1) = 1;
    Peo(i+N,2*i) = 1;
end
P2 = (Pd*Peo)';
```

==

As an example, when N is 4, **P2** is a 8×8 matrix given as

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for} \quad N = 4 \tag{25}$$

P3 is a third permutation matrix, an example of which is generated by the following MATLAB script:

==

```
P3 = zeros(2*N);
for i = 1:N,
    P3(i,2*i-1) = 1;
    P3(N2-i+1,2*i) = 1;
end
```

==

As an example, when N is 4, **P3** is a 8×8matrix given as

$$\mathbf{P3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{for} \quad N = 4 \qquad (26)$$

L1 is a first lifting matrix

$$\mathbf{L1} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z1}_N & \mathbf{I}_N \end{bmatrix} \qquad (27)$$

where $\mathbf{Z1}_N$ is a $N \times N$ counter diagonal matrix given as:

$$\mathbf{Z1}_N = \begin{bmatrix} 0 & 0 & 0 & -\tan(\pi/8N) \\ 0 & 0 & -\tan(3\pi/8N) & 0 \\ 0 & \iddots & 0 & 0 \\ -\tan((2N-1)\pi/8N) & 0 & 0 & 0 \end{bmatrix} \qquad (28)$$

L2 is a second lifting matrix:

$$\mathbf{L2} = \begin{bmatrix} \mathbf{I}_N & \mathbf{Z2}_N \\ \mathbf{O}_N & \mathbf{I}_N \end{bmatrix} \qquad (29)$$

where $\mathbf{Z2}_N$ is a $N \times N$ counter diagonal matrix given as:

$$\mathbf{Z2}_N = \begin{bmatrix} 0 & 0 & 0 & \sin((2N-1)\pi/4N) \\ 0 & 0 & \iddots & 0 \\ 0 & \sin(3\pi/4N) & 0 & 0 \\ \sin(\pi/4N) & 0 & 0 & 0 \end{bmatrix} \qquad (30)$$

L3 is a third lifting matrix:

$$\mathbf{L3} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z3}_N & \mathbf{I}_N \end{bmatrix} \quad (31)$$

where:

$$\mathbf{Z3}_N = \sqrt{2}\mathbf{C}_N^{IV} + \mathbf{I}_N + \mathbf{Z1}_N \quad (32)$$

$\mathbf{L4}$ is a fourth lifting matrix:

$$\mathbf{L4} = \begin{bmatrix} -\mathbf{I}_N & \mathbf{Z4}_N \\ \mathbf{O}_N & \mathbf{I}_N \end{bmatrix} \quad (33)$$

where:

$$\mathbf{Z4}_N = \mathbf{C}_N^{IV} / \sqrt{2} \quad (34)$$

$\mathbf{L5}$ is a fifth lifting matrix:

$$\mathbf{L5} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z5}_N & \mathbf{I}_N \end{bmatrix} \quad (35)$$

where:

$$\mathbf{Z5}_N = -\left(\sqrt{2}\mathbf{C}_N^{IV} + \mathbf{I}_N\right) \quad (36)$$

$\mathbf{L6}$ is a sixth lifting matrix:

$$\mathbf{L6} = \begin{bmatrix} \mathbf{I}_N & \mathbf{O}_N \\ \mathbf{Z6}_N & \mathbf{I}_N \end{bmatrix} \quad (37)$$

where $\mathbf{Z6}_N$ is a $N \times N$ counter diagonal matrix given as:

$$\mathbf{Z6}_N = \begin{bmatrix} 0 & 0 & 0 & \tan(\pi/8) \\ 0 & 0 & \tan(\pi/8) & 0 \\ 0 & \iddots & 0 & 0 \\ \tan(\pi/8) & 0 & 0 & 0 \end{bmatrix} \quad (38)$$

L7 is a seventh lifting matrix:

$$\mathbf{L7} = \begin{bmatrix} \mathbf{I}_N & \mathbf{Z7}_N \\ \mathbf{O}_N & \mathbf{I}_N \end{bmatrix} \quad (39)$$

where $\mathbf{Z7}_N$ is a $N \times N$ counter diagonal matrix given as:

$$\mathbf{Z7}_N = \begin{bmatrix} 0 & 0 & 0 & -\sin(\pi/4) \\ 0 & 0 & \iddots & 0 \\ 0 & -\sin(\pi/4) & 0 & 0 \\ -\sin(\pi/4) & 0 & 0 & 0 \end{bmatrix} \quad (40)$$

L8 is an eighth lifting matrix:

$$\mathbf{L8} = \mathbf{L6} \quad (41)$$

thus, resulting in the factorization as shown in (42):

$$\mathbf{T}_{2N} = \mathbf{P3} \cdot \mathbf{L8} \cdot \mathbf{L7} \cdot \mathbf{L6} \cdot \mathbf{P2} \cdot \mathbf{L5} \cdot \mathbf{L4} \cdot \mathbf{L3} \cdot \mathbf{L2} \cdot \mathbf{L1} \cdot \mathbf{P1} \quad (42)$$

where **P1**, **P2**, and **P3** are three permutation matrices. $\mathbf{L}j$, j from 1 to 8, are eight lifting matrices.

The lifting matrices **L3**, **L4** and **L5** comprise an auxiliary transformation matrix, which is, in this case, the transformation matrix $\mathbf{C}_N^{IV}$ itself.

From Eq. (42), it is possible to compute the integer DCT for two input signals of dimension $N\times 1$.

As Eq. (42) provides a lifting matrix factorization which describes the DCT-IV transformation domain, its lifting matrices can be used in the manner shown herein to compute the domain transformation of an applied input signal.

The equation (42) can be derived in the following way.

The following decomposition can be derived using the disclosure from Wang, Zhongde, "On Computing the Discrete Fourier and Cosine Transforms", IEEE Transactions on Acoustics, Speech and Singal Processing, Vol. ASSP-33, No.4 October 1985:

$$\begin{aligned}\mathbf{C}_N^{IV} &= (\mathbf{B}_N)^T \cdot (\mathbf{P}_N)^T \cdot \begin{bmatrix} \mathbf{C}_{N/2}^{II} & \\ & \overline{\overline{\mathbf{S}_{N/2}^{II}}} \end{bmatrix} \cdot \mathbf{T}_N \\ &= (\mathbf{B}_N)^T \cdot (\mathbf{P}_N)^T \cdot \begin{bmatrix} \mathbf{C}_{N/2}^{II} & \\ & \mathbf{C}_{N/2}^{II} \end{bmatrix} \cdot \mathbf{P}_{DJ} \cdot \mathbf{T}_N\end{aligned} \tag{43}$$

is known, wherein $\mathbf{S}_{N/2}^{II}$ denotes the transformation matrix of the discrete sine transform of type 2,

$$\mathbf{P}_{DJ} = \begin{bmatrix} \mathbf{I} & \\ & \mathbf{D}\cdot\mathbf{J} \end{bmatrix}$$

$\underline{P_N}$ is a $N\times N$ permutation matrix given by

$$\underline{P_N} = \begin{bmatrix} \underline{I_{N/2}} & \\ & \underline{J_{N/2}} \end{bmatrix} \tag{44}$$

$$\underline{T_N} = \begin{bmatrix} \cos\frac{\pi}{4N} & & & & & & & \sin\frac{\pi}{4N} \\ & \cos\frac{3\pi}{4N} & & & & & \sin\frac{3\pi}{4N} & \\ & & \ddots & & & \iddots & & \\ & & & \cos\frac{(N-1)\pi}{4N} & \sin\frac{(N-1)\pi}{4N} & & & \\ & & & -\sin\frac{(N-1)\pi}{4N} & \cos\frac{(N-1)\pi}{4N} & & & \\ & & \iddots & & & \ddots & & \\ & -\sin\frac{3\pi}{4N} & & & & & \cos\frac{3\pi}{4N} & \\ -\sin\frac{\pi}{4N} & & & & & & & \cos\frac{\pi}{4N} \end{bmatrix}$$

and

$$B_N = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2} & & & & & & \\ & 1 & -1 & & & & \\ & 1 & 1 & & & & \\ & & & \ddots & & & \\ & & & & 1 & -1 & \\ & & & & 1 & 1 & \\ & & & & & & \sqrt{2} \end{bmatrix}$$

Equation (85) can be combined with the equation

$$\mathbf{C}_N^{IV} = \mathbf{R}_{PO} \cdot \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \\ & \mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot \mathbf{R}_{PR} \cdot \mathbf{P}_D \cdot \mathbf{P}_{EO} \tag{45}$$

wherein P_{EO} is an even-odd permutation matrix,

$$\underline{R_{pr}} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{N/2} & I_{N/2} \\ \underline{I_{N/2}} & -\underline{I_{N/2}} \end{bmatrix}$$

R_{PO} equals T_N,

$$\mathbf{P}_D = \begin{bmatrix} \mathbf{I}_{\frac{N}{2}} & \\ & \mathbf{D}_{\frac{N}{2}} \end{bmatrix}$$

After transposition equation (45) converts to:

$$\begin{aligned} \mathbf{C}_N^{IV} &= (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T \cdot \mathbf{R}_{PR} \cdot \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \\ & \mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot (\mathbf{R}_{PO})^T \\ &= (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \mathbf{C}_{N/2}^{IV} \\ \mathbf{C}_{N/2}^{IV} & -\mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot (\mathbf{R}_{PO})^T \end{aligned} \tag{46}$$

The combination of (43) and (46) yields:

$$\begin{aligned} \begin{bmatrix} \mathbf{C}_{N/2}^{II} & \\ & \mathbf{C}_{N/2}^{II} \end{bmatrix} &= \mathbf{P}_N \cdot \mathbf{B}_N \cdot (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \mathbf{C}_{N/2}^{IV} \\ \mathbf{C}_{N/2}^{IV} & -\mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot (\mathbf{R}_{PO})^T \cdot \mathbf{T}_N \cdot (\mathbf{P}_{DJ})^T \\ &= \mathbf{P}_3 \cdot \mathbf{R}_2 \cdot \mathbf{P}_2 \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{C}_{N/2}^{IV} & \mathbf{C}_{N/2}^{IV} \\ \mathbf{C}_{N/2}^{IV} & -\mathbf{C}_{N/2}^{IV} \end{bmatrix} \cdot \mathbf{R}_1 \cdot \mathbf{P}_1 \end{aligned} \tag{47}$$

where:

$$\mathbf{P}_1 = (\mathbf{P}_{DJ})^T$$
$$\mathbf{P}_2 = (\mathbf{P}_{EO})^T \cdot (\mathbf{P}_D)^T = (\mathbf{P}_D \cdot \mathbf{P}_{EO})^T$$
$$\mathbf{P}_3 = \mathbf{P}_N$$
$$\mathbf{R}_1 = (\mathbf{R}_{PO})^T \cdot \mathbf{T}_N$$
$$\mathbf{R}_2 = \mathbf{B}_N$$

from (47), equation (42) can be easily derived.

In this embodiment, the computation of the domain transformation requires only 4N rounding operations, as will now be explained:

Let $\alpha(*)$ be the number of real additions, $\mu(*)$ be the number of real multiplications, and $\gamma(*)$ be the number of real roundings, respectively. For the proposed IntDCT algorithm, one gets:

$$\alpha(\text{IntDCT}) = 11N + 3\alpha(\text{DCT-IV})$$
$$\mu(\text{IntDCT}) = 9N + 3\mu(\text{DCT}-\text{IV})$$
$$\gamma(\text{IntDCT}) = 8N$$

The above results are for two blocks of data samples, because the proposed IntDCT algorithm processes them together. Thus for one block of data samples, the numbers of calculations are halved, which are

$$\alpha_1(\text{IntDCT}) = 5.5N + 1.5\alpha(\text{DCT-IV})$$
$$\mu_1(\text{IntDCT}) = 4.5N + 1.5\mu(\text{DCT}-\text{IV})$$
$$\gamma_1(\text{IntDCT}) = 4N$$

where α_1, μ_1, and γ_1 are the number of real additions, number of real multiplications, and number of real roundings, for one block of samples, respectively.

For DCT-IV calculation, the FFT-based algorithm described in the incorporated reference "Signal Processing with lapped Transforms," H. S. Malvar, Norwood, MA. Artech House, 1992, pp. 199-201 can be used, for which

$$\alpha(\text{DCT}-\text{IV}) = 1.5N\log_2 N$$
$$\mu(\text{DCT}-\text{IV}) = 0.5N\log_2 N + N$$

Consequently:

$$\alpha_1(\text{IntDCT}) = 2.25N\log_2 N + 5.5N$$
$$\mu_1(\text{IntDCT}) = 0.75N\log_2 N + 6N$$

In the following, a further embodiment of the method for the transformation of a digital signal from the time domain to the frequency domain and vice versa according to the invention is explained.

In this embodiment a discrete fast fourier transform (FFT) is used as the domain transformation.

Let $\mathbf{F}$ be the $N \times N$ transform matrix of the normalized FFT

$$\mathbf{F} = \sqrt{\frac{1}{N}}\left[\exp\left(\frac{-j2\pi mn}{N}\right)\right]$$
$$m,n = 0,1,\cdots,N-1 \qquad (48)$$

where N is the transform size. m and n are matrix indices.

Under this embodiment, a permutation matrix $\mathbf{P}$ of dimension $N \times N$ is a matrix which includes indices 0 or 1. After multiplying it with a $N \times 1$ vector (the matrix representation of the input signal), the order of elements in the vector are changed.

In this embodiment, lifting matrices are defined as $2N \times 2N$ matrices of the following form:

$$\mathbf{L} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{A} \\ \mathbf{O} & \mathbf{P}_2 \end{bmatrix} \qquad (49)$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are two permutation matrices, $\mathbf{O}$ is the $N \times N$ zero matrix, $\mathbf{A}$ is an arbitrary $N \times N$ matrix. For lifting matrix $\mathbf{L}$, reversible integer to integer mapping is realized in the same way as the 2×2 lifting step in the aforementioned incorporated reference of I. Daubechies. As above, however, rounding is applied to a vector instead of a single variable. It is apparent that the transposition of $\mathbf{L}$, $\mathbf{L}^T$ is also a lifting matrix.

Further, let $\mathbf{T}$ be a $2N \times 2N$ transform matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{O} & \mathbf{F} \\ \mathbf{F} & \mathbf{O} \end{bmatrix} \qquad (50)$$

Accordingly, the modified transform matrix $\mathbf{T}$ (and accordingly the domain transformation itself) can be expressed as the lifting matrix factorization:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ -\mathbf{Q}\cdot\mathbf{F} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} -\mathbf{Q} & \mathbf{F} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{F} & \mathbf{I} \end{bmatrix} \qquad (51)$$

where $\mathbf{I}$ is the $N \times N$ identity matrix, and $\mathbf{Q}$ is a $N \times N$ permutation matrix given as:

$$\mathbf{Q} = \begin{bmatrix} 1 & \mathbf{O}_{1\times N-1} \\ \mathbf{O}_{N-1\times 1} & \mathbf{J} \end{bmatrix} \qquad (52)$$

and $\mathbf{O}_{1\times N-1}$ and $\mathbf{O}_{N-1\times 1}$ are row and column vectors of $N-1$ zeros respectively.

$\mathbf{J}$ is the $(N-1)\times(N-1)$ counter index matrix given by

$$\mathbf{J}=\begin{bmatrix} & & & 1 \\ & & 1 & \\ & \cdot^{\cdot^{\cdot}} & & \\ 1 & & & \end{bmatrix} \tag{53}$$

In Eq. (53), blank space in the square bracket represents all zeros matrix elements.

A can be seen from Eq. (51), the lifting matrix factorization can be use to compute the integer FFT for two $N\times 1$ complex vectors using the methods as described herein.

Under this embodiment, the computation of the domain transformation requires only 3N rounding operations, as will now be explained:

Let: $\alpha(*)$ be the number of real additions,

$\mu(*)$ be the number of real multiplications, and

$\gamma(*)$ be the number of real rounding operations,

respectively.

For the proposed IntFFT algorithm, we have

$$\alpha(\text{IntFFT})=6N+3\alpha(\text{FFT})$$

$$\mu(\text{IntFFT}) = 3\mu(\text{FFT})$$
$$\gamma(\text{IntFFT}) = 6N$$

The above results are for two blocks of data samples, because the proposed IntFFT algorithm processes them together. Thus for one block of data samples, the numbers of calculations are halved, which are

$$\alpha_1(\text{IntFFT}) = 3N + 1.5\alpha(\text{FFT})$$
$$\mu_1(\text{IntFFT}) = 1.5\mu(\text{FFT})$$
$$\gamma_1(\text{IntFFT}) = 3N$$

where α_1, μ_1, and γ_1 are the number of real additions, number of real multiplications, and number of real rounding operations for one block of samples, respectively.

For FFT calculation, the split-radix FFT (SRFFT) algorithm can be used, for which:

$$\alpha(\text{SRFFT}) = 3N\log_2 N - 3N + 4$$
$$\mu(\text{SRFFT}) = N\log_2 N - 3N + 4$$

Consequently, we have:

$$\alpha_1(\text{IntFFT}) = 4.5N\log_2 N - 1.5N + 6$$
$$\mu_1(\text{IntFFT}) = 1.5N\log_2 N - 4.5N + 6$$

Figure 7 shows forward and reverse transform coders used to assess the transform accuracy of the DCT transformation technique described above and the FFT domain transformation above. The test involved measuring the mean squared error (MSE) of the transform in accordance with the evaluation standards proposed by the MPEG-4 lossless audio coding group as described in "Coding of Moving Pictures and Audio: Work plan for Evaluation of Integer MDCT for FGS to Lossless

Experimentation Framework," ISO/IEC JTC 1/SC 29/WG 11 N5578 Pattaya, Thailand, Mar. 2003 incorporated herein.

Specifically, the MSEs for IntDCT and integer inverse DCT (IntIDCT) are given as:

$$MSE = \frac{1}{K}\sum_{j=0}^{K-1}\frac{1}{N}\sum_{i=0}^{N-1}e_i^2 \tag{54}$$

where the error signal e is e_f for IntDCT, and e_i for IntIDCT as in Fig 1. K is the total number of sample blocks used in the evaluation.

The MSEs for IntFFT and integer inverse FFT (IntIFFT) are given as

$$MSE = \frac{1}{K}\sum_{j=0}^{K-1}\frac{1}{N}\sum_{i=0}^{N-1}\|e_i\|^2 \tag{55}$$

where the error signal e is e_f for IntFFT, and e_i for IntIFFT as in Fig 1. $\|*\|$ represents norm of a complex value. K is the total number of sample blocks used in the evaluation.

For both domain transformations, a total of 450 seconds with 15 different types of music files are used in the 48 kHz/16-bit test set. Table I shows the test results.

As can be seen from Table 1, the MSE generated using the systems and methods of the present invention is very minimal, and unlike conventional systems, is substantially independent of the processing block size. Referring to the DCT-IV domain

transformation, the MSE only slightly increases with increasing block size N up to 4096 bits. The MSEs of the FFT are even better, exhibiting a constant MSE of 0.4 for block sizes up to 4096 bits. When the demonstrated performance of the present invention is viewed in light of the present capabilities and increasing need for longer block sizes, the advantages of the present invention become clear.

N	IntDCT-IV	IntIDCT-IV	IntFFT	IntIFFT
8	0.537	0.537	0.456	0.371
16	0.546	0.546	0.480	0.412
32	0.549	0.548	0.461	0.391
64	0.550	0.550	0.462	0.393
128	0.551	0.551	0.461	0.391
256	0.552	0.552	0.461	0.391
512	0.552	0.552	0.461	0.391
1024	0.552	0.552	0.460	0.391
2048	0.552	0.552	0.461	0.391
4096	0.553	0.552	0.461	0.391

Table I

Incorporated References

The following documents are herein incorporated by reference:

H. S. Malvar, "Signal Processing with Lapped Transforms" *Artech House*, 1992;

R. Geiger, T. Sporer, J. Koller, K. Brandenburg, "Audio Coding based on Integer Transforms" *AES 111th Convention*, New York, USA, Sept. 2001;

Wang, Zhongde, "On Computing the Discrete Fourier and Cosine Transforms", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-33, No.4 October 1985;

I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps", *Tech. Report*, Bell Laboratories, Lucent Technologies, 1996;

S. Oraintara, Y. J. Chen and T. Q. Nguyen, "Integer fast Fourier transform", *IEEE Trans. Signal Processing*, vol. 50, no. 3, Mar. 2002, pp. 607-618;

P. Hao and Q. Shi, "Matrix factorizations for reversible integer mapping," *IEEE Trans. Signal Processing*, vol. 49, no. 10, Oct. 2001, pp. 2314-2324;

G. Plonka and M. Tasche, "Invertible integer DCT algorithms", *Appl. Comput. Harmon. Anal.* 15:70-88, 2003;

Y. H. Zeng, L.Z. Cheng, G. A. Bi, and Alex C. Kot, "Integer DCTs and fast algorithms", *IEEE Trans. Signal Processing*, vol. 49, no. 11, Nov. 2001, pp. 2774-2782;

J. Wang, J. Sun and S. Yu, "1-D and 2-D transforms from integers to integers", in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Hong Kong, 2003, vol. II, pp. 549-552;

"Coding of Moving Pictures and Audio: Work plan for Evaluation of Integer MDCT for FGS to Lossless Experimentation Framework", ISO/IEC JTC 1/SC 29/WG 11 N5578, Pattaya, Thailand, Mar. 2003.